Amendments to the Claims:

Please amend claims 1 -7, 9 -18, 20 – 28, 32 - 51 and 58, cancel claims 19, 29 – 31, and 52 – 54, and add new claims 59 - 65 as noted in the listing below. This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method, comprising:

~~receiving~~ encountering a non-privileged user-level programming instruction[[s]] ~~to execute one or more shared resource threads (shreds) via an instruction set architecture (ISA)~~;

~~configuring one or more instruction sequencers via the ISA; and~~

creating, responsive to the programming instruction, a first shared resource thread (shred) that shares virtual memory address space with one or more other shreds; and

executing, responsive to the programming instruction, the shred concurrently with at least one of the one or more other shreds~~; simultaneously with a microprocessor that includes multiple instruction sequencers~~

wherein creating the shred is performed in hardware, without the intervention of an operating system.

2. (Currently Amended) The method of claim 1, ~~wherein the ISA provides~~ further comprising:

maintaining a private state for the first thread, ~~each shred of the one or more shreds, and~~ wherein the private state is associated with at least one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register.

3. (Currently Amended) The method of claim 1, further comprising:

sharing a state among a plurality of shreds associated with a first thread, the plurality of shreds including the first shred and the one or more other shreds; ~~of the one or more shreds, while maintaining the state privately among one shred of the plurality of shreds, wherein the state shared is associated with one of a plurality of registers including a control register, a flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers~~

while not sharing said state with a second shred that is associated with a second thread.

4. (Original) The method of claim 1, further comprising:

sharing a state among a plurality of shreds of the one or more shreds; and

storing the state in one or more registers.

5. (Currently Amended) The method of claim 1, wherein the first shred and the one or more shreds share a current privilege level and share a common address translation.

6. (Currently Amended) The method of claim 1, further comprising:

receiving a non-privileged user-level programming instruction that encodes a shred ~~wherein the ISA includes instructions to create, and~~ destroy operation~~the one or more shreds~~.

7. (Currently Amended) The method of claim 1, further comprising communicating between the first shred and at least one of the one or more other shreds~~via one or more shared registers~~.

8. (Original) The method of claim 1, further comprising sharing a system state among the one or more shreds.

9. (Currently Amended) The method of claim [[1]]7, wherein said communicating is performed via ~~the one or more shreds determine remote and local relationships to each shred of the~~ one or more [[shreds]]shared registers.

10. (Currently Amended) The method of claim 1, further comprising:

~~associating the one or more shreds with a thread;~~

scheduling, responsive to a user-level programming instruction, ~~wherein an application program controls~~ the first shred and the one or more other shreds without intervention of the operating system~~directly, including scheduling of the one or more shreds, and~~

~~wherein an operating system executed by the multiprocessor schedules one or more threads~~.

11. (Currently Amended) The method of claim [[1]]7, wherein~~further comprising~~:

~~associating the one or more shreds with a thread; and~~

~~suspending the one or more shreds belonging to the thread when a context switch request is received through a single shred of the one or more shreds.~~ said communicating is performed via a user-level shred signaling instruction.

12. (Currently Amended) The method of claim 11, further comprising:

storing one or more shred states associated with the one or more shreds responsive to receipt of a ~~when the~~ context switch request~~ is received~~.

13. (Currently Amended) The method of claim 1, further comprising:

~~a first process reporting one or more exceptions; and~~

~~a second process servicing the one or more exceptions~~

handling with user-level exception handler code an exception generated during execution of the first shred, without intervention of the operating system.

14. (Currently Amended) The method of claim 13, further comprising:

receiving the ~~one or more exceptions~~ exception from an application program; and

determining whether to report the ~~one or more exceptions~~ exception to the operating system.

15. (Currently Amended) The method of claim [[14]]1, wherein~~further comprising~~:

~~prioritized reporting of the one or more exceptions and one or more context changes to the operating system; comprising~~

~~receiving the one or more exceptions simultaneously via different shreds of the one or more shreds; and~~

~~servicing one of the one or more exceptions and the one or more context changes according to the prioritized reporting while suspending exception processing of other exceptions of the one or more exceptions~~.

the shred is to perform input/output (I/O) operations.

16. (Currently amended) The method of claim 1, wherein the one or more shreds are to perform ~~input/output (I/O) functions and~~ computation functions.

17. (Currently Amended) An apparatus, comprising:

execution resources to execute a plurality of instructions, the execution resources including ~~a microprocessor that includes~~ multiple instruction sequencers;

the execution resources to receive a non-privileged user instruction;

the execution resources further to, responsive to the received instruction, begin execution of a shred concurrently with one or more other shreds. ~~and~~

~~a plurality of user-level multithreading registers coupled to the microprocessor that~~

~~receives programming instructions to execute one or more shreds in accordance with an instruction set architecture (ISA);~~

~~configures one or more instruction sequencers via the ISA; and~~

~~executes the one or more shreds simultaneously~~.

18. (Currently Amended) The apparatus of claim 17, further comprising: ~~wherein the plurality of user-level multithreading registers further comprises a plurality of~~ one or more shared shred registers to facilitate communication between ~~a plurality~~ two or more of the shreds ~~and to facilitate synchronization between the plurality of shreds~~.

19. Cancelled.

20. (Currently Amended) The apparatus of claim [[17]]18, wherein the ~~plurality of user-level multithreading~~ one or more shared registers further comprise ~~comprises~~ a first register that enables an operating system or BIOS to enable multithreading architecture extensions for user-level multithreading.

21. (Currently Amended) The apparatus of claim 17, wherein the execution resources are further to, responsive to the received instruction, begin execution of a shred concurrently with one or more other shreds, without control of an operating system. ~~ISA allows a private state for each shred of the one or more shreds, and wherein the private state is associated with one of a plurality of registers including general-purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register~~.

22. (Currently Amended) The apparatus of claim 17, wherein the execution resources include one or more processor cores capable of executing multiple shreds concurrently. ~~ISA allows sharing a state among a plurality of shreds of the one or more shreds, while maintaining the state privately among one shred of the plurality of shreds, wherein the state shared is associated with one of a plurality of registers including a control register, a flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers.~~

23. (Currently Amended) The apparatus of claim 17, further comprising:
One or more registers to hold a state shared ~~sharing a state~~ among the shred and ~~a plurality of shreds of~~ the one or more other shreds~~; and storing the state in the plurality of user-level multithreading registers~~.

24. (Currently Amended) The apparatus of claim 17, wherein the shred and the one or more other shreds share a current privilege level and share a common address translation.

25. (Currently Amended) The apparatus of claim 17, further comprising logic to execute a user-level instruction to create the shred. ~~wherein the ISA includes instructions to create, and destroy the one or more shreds~~.

26. (Currently Amended) The apparatus of claim 17, further comprising a mechanism to perform communication ~~communicating~~ between the shred

and the one or more other ~~shreds via the plurality of user-level multithreading registers~~.

27. (Currently Amended) The apparatus of claim 17, further comprising sharing a system state among the shred and the one or more other shreds.

28. (Currently Amended) The apparatus of claim [[17]]26, wherein the mechanism further comprises one or more shared registers~~one or more shreds determine remote and local relationships to each shred of the one or more shreds~~.

29. Cancelled.

30. Cancelled.

31. Cancelled.

32. (Currently Amended) The apparatus of claim 17, further comprising ~~wherein the ISA~~:

a user-level exception mechanism to report an exception to the shred.

~~_reports one or more exceptions by a first process; and~~

~~services the one or more exceptions by a second process.~~

33. (Currently Amended) The apparatus of claim [[32]]17, further comprising ~~the microprocessor~~:

an exception mechanism ~~receiving the one or more exceptions from an application program; and determines whether~~ to report ~~the one or more exceptions~~ an exception to [[the]] an operating system.

34. (Currently Amended) The apparatus of claim [[33]]32, further comprising: ~~wherein the ISA~~

a mechanism to detect multiple exceptions, each exception associated with a different one of a plurality of concurrently-executing shreds, where the plurality includes the shred and the one or more other shreds;

wherein the exception mechanism includes a prioritizer to prioritize the exceptions;

and wherein the exception mechanism is further to report only one of the prioritized exceptions at a time to the operating system.

~~permits prioritized reporting of the one or more exceptions and one or more context changes to the operating system upon receiving the one or more exceptions simultaneously via different shreds of the one or more shreds; and~~

~~services one of the one or more exceptions and the one or more context changes according to the prioritized reporting while suspending exception processing of other exceptions of the one or more exceptions.~~

35. (Currently Amended) An article of manufacture, comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to perform operations comprising,

receiving user-level programming instructions to execute ~~one or more~~ a plurality of shared resource threads (shreds) ~~via an instruction set architecture (ISA)~~;

configuring one or more instruction sequencers responsive to the one or more user-level programming instructions ~~via the ISA~~;

scheduling the shreds via hardware; and

executing the ~~one or more~~ plurality of shreds concurrently ~~simultaneously with a microprocessor that includes~~ on multiple instruction sequencers.

36. (Currently Amended) The article of manufacture of claim 35, wherein the operations further comprise:

~~ISA provides~~ maintaining a private state for each shred of the ~~one or more~~ plurality of shreds, ~~and~~ wherein the private state is associated with at least one of a plurality of registers including general purpose registers, floating point registers, MMX registers, segment registers, a flags register, an instruction pointer, control and status registers, SSE registers, and a MXCSR register.

37. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising sharing a first state among [[a]] the plurality of shreds ~~of the one or more shreds~~, while maintaining [[the]] a second state privately among an additional [[one]] shred associated with a thread that is not associated with the plurality of shreds ~~of the plurality of shreds~~, wherein the first state [[shared]] is associated with at least one of a plurality of registers including a control register, a

flags register, memory management registers, a local descriptor table register, a task register, debug registers, model specific registers, shared registers, and shred control registers.

38. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

sharing a state among [[a]] the plurality of shreds ~~of the one or more shreds~~; and storing the state in one or more registers.

39. (Currently Amended) The article of manufacture of claim 35, wherein the plurality of ~~one or more~~ shreds share a current privilege level and share a common address translation.

40. (Currently Amended) The article of manufacture of claim 35, wherein the one or more user-level programming instructions ~~ISA includes~~ include an instruction ~~instructions~~ to create~~, and destroy~~ ~~the~~ one or more of the plurality of shreds.

41. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising communicating [[between]] among the ~~one or more~~ plurality of shreds ~~via one or more shared registers~~.

42. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising sharing a system state among the plurality of ~~one or more~~ shreds.

43. (Currently Amended) The article of manufacture of claim 35, wherein ~~the one or more shreds determine remote and local relationships to each shred of the one or more shreds~~ the machine-accessible medium further includes data that causes the machine to perform operations comprising communicating between the plurality of shreds via one or more shared registers.

44. (Currently Amended) The article of manufacture of claim 35, wherein an application program controls the plurality of ~~one or more~~ shreds directly, including scheduling of the ~~one or more~~ plurality of shreds, and wherein an operating system executed by the multiprocessor schedules one or more threads.

45. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

associating the ~~one or more~~ plurality of shreds with a thread; and

suspending the ~~one or more~~ plurality of shreds belonging to the thread when a context switch request is received through a single one of ~~shred of~~ the ~~one or more~~ plurality of shreds.

46. (Currently Amended) The article of manufacture of claim 45, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

storing one or more shred states associated with the plurality of ~~one or more~~ shreds when the context switch request is received.

47. (Currently Amended) The article of manufacture of claim 35, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

~~a first process~~ reporting one or more exceptions to a first shred of the plurality of shreds~~; and~~

~~a second process servicing the one or more exceptions~~.

48. (Currently Amended) The article of manufacture of claim 47, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

[[receiving]] reporting the one or more exceptions from an application program; and

determining whether to report the one or more exceptions to [[the]] an operating system.

49. (Currently Amended) The article of manufacture of claim 48, wherein the machine-accessible medium further includes data that [[cause]] causes the machine to perform operations comprising:

prioritized-reporting of the one or more exceptions ~~and one or more context changes~~ to the operating system; comprising

receiving the one or more exceptions [[simultaneously]] concurrently via different shreds of the ~~one or more~~ plurality of shreds; and

servicing one of the one or more exceptions ~~and the one or more context changes~~ according to the prioritized-reporting while suspending exception processing of other exceptions of the one or more exceptions.

50. (Currently Amended) The article of manufacture of claim 35, wherein the [[one or more]] plurality of shreds perform input/output (I/O) functions and computation functions.

51. (Currently Amended) A system, comprising:

a microprocessor implementing an instruction set architecture (ISA), the microprocessor capable of executing multiple concurrent shreds; and

a memory;

~~a plurality of user-level multithreading registers coupled to the microprocessor, wherein the plurality of user-level multithreading registers comprises a first register that enables an operating system or BIOS to enable multithreading architecture extensions for user-level multithreading; and~~

~~an instruction set architecture (ISA) for a 64-bit architecture or 32-bit architecture compatible with the microprocessor and the plurality of~~ wherein the ISA includes one or more instructions to allow user-level multithreading [[registers] operations.

52. Cancelled.

53. Cancelled.

54. Cancelled.

55. (Original) A system, comprising:

a microprocessor, including

a plurality of user-level multithreading registers coupled to the microprocessor; and

memory coupled to the microprocessor that stores an instruction set architecture (ISA) compatible with the microprocessor and the plurality of user-level multithreading registers, wherein the memory is from a plurality of memory devices including DRAM, flash, and EEPROM,

wherein the plurality of user-level multithreading registers and the ISA enable multithreading architecture extensions for user-level multithreading.

56. (Original) The system of claim 55, wherein the plurality of user-level multithreading registers further comprises a plurality of shared shred registers to facilitate communication between a plurality of shreds and to facilitate synchronization between the plurality of shreds.

57. (Original) The system of claim 56, wherein the plurality of user-level multithreading registers further comprises a plurality of shred control registers to manage the plurality of shreds.

58. (Currently Amended) The system of claim 57, wherein the microprocessor:

receives programming instructions to execute one or more shreds in accordance with the ISA;

configures one or more instruction sequencers via the ISA; and

executes the one or more shreds [[simultaneously]] concurrently.

59. (New) The apparatus of claim 32, wherein:

the user-level exception mechanism is further to vector to a fixed location in order to allow the shred to service to the exception.

60. (New) The apparatus of claim 32, wherein:

the plurality of instructions further include a system call instruction to explicitly invoke an operating system to service to the exception.

61 (New) The apparatus of claim 33, wherein said prioritizer employs a round-robin scheme.

62. (New) The article of manufacture of claim 35, wherein the one or more user-level programming instructions include an instruction to destroy one or more of the plurality of shreds.

63. (New) The system of claim 51, wherein the one or more instructions include an instruction to create a shred without intervention of an operating system.

64. (New) The system of claim 51, wherein the one or more instructions include an instruction to destroy a shred without intervention of an operating system.

65. (New) The system of claim 51, wherein:

the user-level multi-threading operations include concurrent execution of two or more shreds associated with the same thread.